

# **Prosjektrapport**

Ekspert i Team - Robot og menneske

## **Implementering og automatisering av utskytningsvinkel på fotballmaskin**



VÅR 2015

### **Gruppe 5**

Brage Iversen, Erik Karlsson, Martinius Knudsen, Eirik Knævelsrud,  
Sondre Landvik, Magnus Rammel, Frode André Sundal

## Sammendrag

Oppgaven tar utgangspunkt i en tidligere masteroppgave ved institutt for kybernetikk. Denne har lagt til rette for styring og beregning av ballbane for en enkel fotballmaskin for keeper- og spillertrening. Maskinen er av typen Soccer Tutor Pro Trainer og laget av Sports Tutor Inc.

Det har vært et ønske om å videreutvikle denne maskinen med automatisering og styringssystem for to nye akser. Fra før kunne maskinen stilles inn i en gitt posisjon, samt låses manuelt i en gitt utskytningsvinkel. Utskytningshastighet og tid mellom hvert skudd kunne stilles med originalt styringssystem, men ikke fra en og samme enhet. I masteroppgaven var det videreutviklet en siktemekanisme slik at en kunne stille inn en gitt avstand som ballen treffer med en presisjon på én kvadratmeter.

Vi har anvendt produktutviklingsmetodikk for å undersøke muligheter og konsepter for motorisering, samt mekaniske aspekter. Kode fra tidligere oppgave er analysert, modifisert og utbedret for å kunne kjøres med en nyere versjon av programmeringsspråket Python og MATLAB. Det er også utviklet en mobilapplikasjon for å legge til rette for brukervennlig styring av hele systemet, med alle funksjoner implementert.

Grunnet utfordringer mtp. tid og ressurser, samt andre komplikasjoner, har gruppen kun fått implementert styring for vertikal tilt. For horisontal rotasjon er det kun gjort et konseptstudie, der gruppen har tenkt ut teoretiske forslag til løsninger. Konsepttegninger for dette inngår i rapporten. I tillegg har gruppen også samlet alle funksjoner til å styres fra en enhet.

Gruppen har jobbet godt med oppgaven fra dag 1, men det ble oppdaget tidlig at arbeidet var av et betydelig større omfang enn først antatt. Kode fra eksisterende oppgave var langt fra å være "plug and play". Valg av motor og girløsning var også svært tidkrevende, da det var vanskelig å finne de spesifikasjoner som var nødvendige, og samtidig holde seg til et stramt budsjett.

Sluttresultatet har blitt mindre omfattende enn håpet, men arbeidet er gjort grundig, og det er lagt et meget godt grunnlag for videre utvikling.

# Innhold

Sammendrag .....	1
Figurliste .....	3
Tabelliste .....	3
Etablering av begreper .....	4
1 Innledning.....	5
1.1 Bakgrunn for oppgaven .....	5
1.2 Motivasjon.....	5
1.3 Begrensninger.....	5
1.4 Bidrag.....	6
1.5 Disposisjon.....	6
2 Konseptstudie.....	7
2.1 Masteroppgaven .....	7
2.2 Produktutvikling .....	7
2.2.1 Andre løsninger .....	7
2.2.2 Vertikal tilt .....	8
2.2.3 Horisontal rotasjon.....	13
2.3 Digitalt .....	14
2.3.1 Styringskrets .....	14
2.3.2 Komponenter.....	14
2.3.3 Feedback på tiltvinkel.....	15
2.3.4 Horisontal styring .....	15
2.3.5 Kommunikasjon med bruker .....	16
3 Beregninger .....	17
3.1 Moment, rotasjonshastighet og giring. ....	17
3.2 CAD og simuleringer .....	17
3.3 Produksjon.....	18
4 Løsning.....	19
4.1 Mekanisk system .....	19
4.2 Mekatronisk/elektronisk system.....	19
5 Implementering.....	21
5.1 Mekanisk .....	21
5.2 Digitalt (Elektronikk og programvare) .....	21
5.2.1 Styring.....	22
5.2.2 Oppsett .....	24
5.2.3 Programvare .....	25
5.2.4 Mobilapplikasjon .....	26
6 Vår uttesting/resultater .....	29
6.1 Initiell testing.....	29

6.2	Fortløpende testing under utvikling .....	29
6.3	Testing av girløsning .....	29
6.4	Testing av sluttresultat .....	30
6.5	Mobilapplikasjon .....	30
7	Diskusjon .....	31
8	Konklusjon .....	33
9	Videre arbeid .....	34

## Figurliste

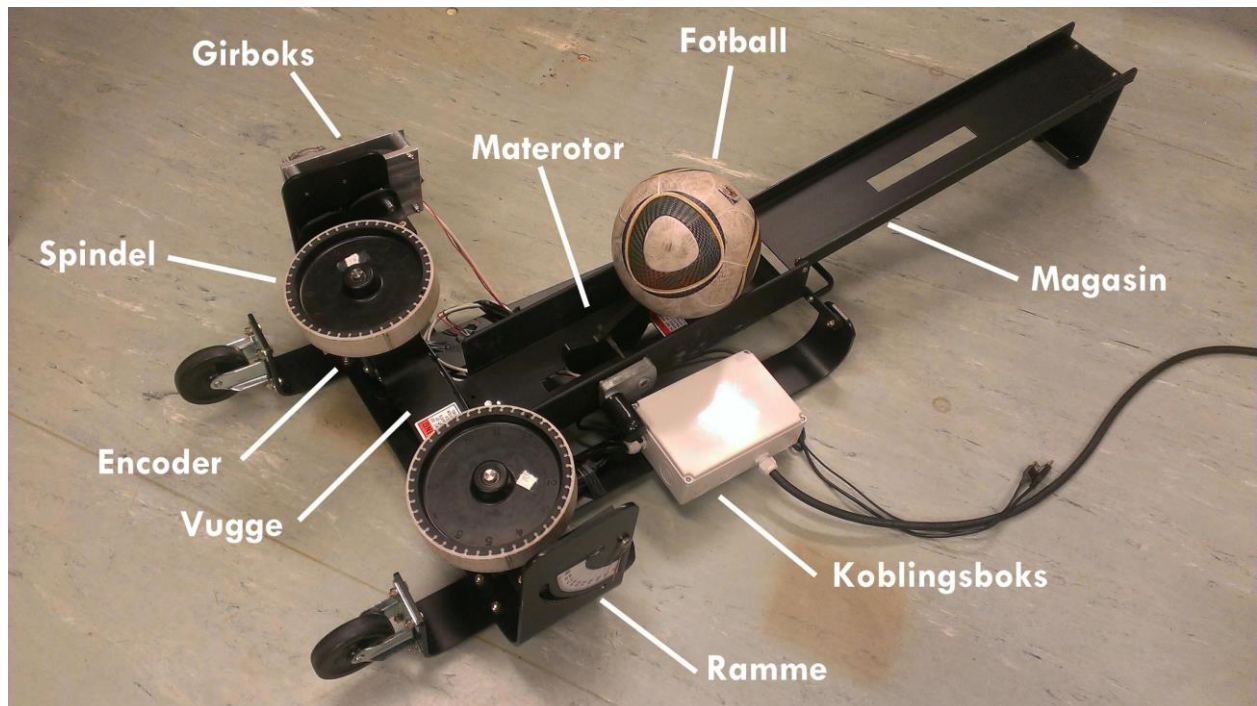
Figur 1: Fotballmaskin med navn på komponenter .....	4
Figur 2: Andre eksisterende løsninger .....	7
Figur 3: Alternative teknologier .....	8
Figur 4: Overordnet styringskrets .....	14
Figur 5: 3D-modell av fotballmaskin .....	17
Figur 6: 3D-modell av girkasse med motor .....	18
Figur 7: Girboks .....	19
Figur 8: Overordnet styringskrets, valgt løsning .....	20
Figur 9: Ferdig montert girboks .....	21
Figur 10: Styringskrets med kommunikasjonsforløp .....	22
Figur 11: Arduino mikrokontroller .....	22
Figur 12: Driverkrets .....	23
Figur 13: Koblingsboks .....	24
Figur 14: Roboclaw motorkontroller .....	24
Figur 15: Klassesdiagram .....	25
Figur 16: Brukergrensesnitt .....	26
Figur 17: Kommunikasjon mellom app og kontroller .....	28

## Tabelliste

Tabell 1: Konseptutvikling, plassering av rotasjonsakse og motor .....	10
Tabell 2: Motorsammenligning .....	11
Tabell 3: Girprinsipper .....	12
Tabell 4: Prinsipper for horisontalbevegelse .....	13
Tabell 5: Roboclaw konfigurasjon. Ledningsoppsett for kommunikasjon med Roboclaw .....	24
Tabell 6: Arduino pin-konfigurasjon .....	25
Tabell 7: Grensesnitt i mobilapplikasjon .....	27

## Etablering av begreper

GUI	Graphical User Interface
RPM	Revolutions Per Minute
PID	Proporsjonal Integrasjon Derivasjon
Kp, Ki, Kd	PID-konstanter for regulering
DC	Direct Current
IR	Infrared
CAD	Computer Aided Design
CAE	Computer Aided Engineering
FEM	Finite Element Method
MOSFET	Metal–Oxide–Semiconductor Field-Effect Transistor
TCP	Transmission Control Protocol



Figur 1: Fotballmaskin med navn på komponenter

# 1 INNLEDNING

## 1.1 BAKGRUNN FOR OPPGAVEN

Maskinen fungerte i prinsipp på samme måte som i dag da vi fikk den. Man legger fotballen i magasinet Figur 1: Fotballmaskin med navn på komponenter og materotoren slipper ballene frem til spindlene med ønsket intervall. Ballen triller så ned til spindlene som griper tak i ballen og skyter den ut. Opprinnelig ble utskytningsvinkel stilt inn manuelt og med liten presisjon. Grunnet mye friksjon i eksisterende løsning var dette også unødvendig tungt. Ballmater hastighet ble styrt via en kontrollboks på siden av maskinen, slik den var satt opp fra fabrikken. Spindlene ble styrt via PC fra det masteroppgaven hadde implementert. Maskinen kan pakkes sammen ved å folde inn magasinet, og kan dermed lett transporteres på hjulene montert foran.

I masteroppgaven det er tatt utgangspunkt i er det laget MATLAB-kode for utregning av ballbanen, kontrollert av utgangshastighet og utgangsvinkel. Dette danner grunnlag for styringen av spindlene, som kontrolleres gjennom programvaren og en motorkontroller. Denne motorkontrolleren gjør det mulig å styre spindlene uavhengig av hverandre. Frem til nå er det kun hastighet som er automatisert, selv om de andre parameterne regnes ut i programmet. Dette gjør at maskinen fortsatt er avhengig av direkte menneskelig interaksjon. Det er derfor ønskelig å fullføre automasjonen av utskytingen, samt å kunne fjernstyre maskinen trådløst fra f.eks. en smarttelefon.

## 1.2 MOTIVASJON

Opgaven ble valgt ut blant de foreslåtte oppgavene i landsbyen. Motivasjonen for å velge den var i hovedsak tanken på å kunne jobbe med et praktisk prosjekt, noe som var av interesse for samtlige gruppemedlemmer. Det å kunne ta utgangspunkt i et eksisterende produkt og videreutvikle dette, fremsto som et godt grunnlag for et prosjekt med begrenset tid.

Videreutvikling og automatisering av en fotballmaskin kunne åpne for interessante muligheter innen trening av fotballspillere. Gruppen så for seg at en slik maskin potensielt kunne effektivisere og forbedre keepertrening, med blant annet en rekke forskjellige skudd med endring i både styrke, vinkel og spinn. Maskinen kan også bli brukt til spillertrening for blant annet corner-trening, innkastsimulering og generell trening på mottak av ball, der konsistente skudd kan være fordelaktige.

## 1.3 BEGRENSNINGER

Både tid og penger har vært en betydelig utfordring for gruppen i dette prosjektet. Motorer med de dimensjoner og egenskaper som gruppen syntes var nødvendig for å styre denne maskinen er forbundet med store kostnader. Gruppen så ingen mulighet for å holde seg innenfor den eksisterende budsjettrammen på 1000 kr. Ettersom dette prosjektet både var blant de foreslåtte prosjektoppgavene for hele landsbyen, og var støttet opp av faglærers vilje til å se det gjennomført, fikk gruppen bevilget et større budsjett til innkjøp av alt de trengte. Horisontal rotasjon ble forkastet

til fordel for å kunne lage best mulig løsning for vertikal tilt. Det ble også gjort en del kompromisser på motor- og girløsning for vertikal tilt for å holde kostnader nede. Kvaliteten er likevel blitt god.

## 1.4 BIDRAG

En motor- og girløsning har blitt montert på vuggen og tillater dermed automatisk justering av tiltvinkelen. Styring av dette og resten av maskinen utføres av en egenprodusert H-bridge og et motorkontrollkort, og for brukervennlighet er det også utviklet en mobilapplikasjon for å tillate fjernstyring av hele systemet. Alt er samlet i et felles brukergrensesnitt for å enkelt kunne kontrollere alle funksjonene på maskinen som ballskyterhastighet, ballmaterhastighet og vinkelstyring.

## 1.5 DISPOSISJON

Rapporten presenterer først arbeidet gjennomført i forbindelse med. konseptstudie og valg av løsninger, før implementering og testing av hele systemet. Noe teori og diskusjon er også inkludert for å tydeliggjøre arbeidet som er gjennomført, men det overordnede fokuset i prosjektet har vært på å implementere de fysiske og digitale løsningene.

## 2 KONSEPTSTUDIE

Da denne oppgaven tar utgangspunkt i videreutvikling av et produkt har vi gått inn for å anvende produktutviklingsmetodikk for å undersøke løsningsrom og muligheter.

### 2.1 MASTEROPPGAVEN

Gruppen fikk tilgang til masteroppgaven som var skrevet rundt maskinen og begynte å sette seg inn i dens innhold. Det skulle raskt vise seg at dagens løsning ikke var “plug and play”, da koden for å kjøre programmets funksjoner var fordelt over mange enkeltfiler og biblioteker. Bibliotekene som var nødvendig for å kjøre viste seg å ikke være listet opp i oppgaven. Det ble derfor gjort arbeid med å samle alt av kode, biblioteker og omprogrammere fra Linux-oppsett for å kunne kjøre maskinen med program og funksjoner fra denne oppgaven. I tillegg var heller ikke oppsett av maskinen, mtp. ledningsoppsett og protokoll for tilkobling av strøm til systemet, beskrevet slik det burde vært for å forenkle prosessen med å sette opp systemet.

### 2.2 PRODUKTUTVIKLING

#### 2.2.1 Andre løsninger

Opprinnelig maskin kommer fra den amerikanske produsenten Sports Tutor Inc. De har størst fokus på baseball- og tennismaskiner, men har også brukt lignende teknologi til å utvikle volleyball- og fotballmaskiner. Det ble derfor undersøkt hvorvidt disse løsningene hadde mer omfattende styring og frihetsgrader enn maskinen gruppen skulle jobbe med. Maskinene laget for større prosjektiler er enklere, har manuell styring, og ofte manuell mating. De mindre maskinene er betydelig mer omfattende med både større magasiner og automatisk styring av 2 akser, samt mulighet for skru på enkelte modeller.

Det ble funnet andre produsenter med potensielt bedre løsninger. Tilsynelatende også mer moderne. Eksempler som *Sidekick soccer machine* har spindlene plassert på en 3-akset rigg på en liten plattform med 4 hjul. Dette ser ut til å være den mest avanserte fotballmaskinen på markedet når det kommer til mulighet for variasjon i kulebaner. Den har dog ikke magasin, og er avhengig av å bli styrt og matet av en person. Alternativer kan være *Toca machine* som har et elegant og kompakt magasin, men det er usikkerheter rundt kompleksiteten til styringssystemet.



Figur 2: Andre eksisterende løsninger



## *Alternative teknologier*

Det ble også gjort undersøkelser av alternative teknologier og løsninger som kunne kombineres på nye måter for å løse utfordringen med akseimplementering. Ettersom det var behov for litt kraft grunnet maskinens størrelse og vekt, ble det undersøkt større eksisterende systemer som våpenturret, lyskastere og vinklingssystemer for solcellepaneler.



**Figur 3: Alternative teknologier**

### 2.2.2 Vertikal tilt

Det ble tidlig oppdaget at dagens løsning har et betydelig “offset” mellom vuggens tyngdepunkt og rotasjonsakse. Det var derfor ønskelig å se på muligheten for å flytte rotasjonsaksen for å redusere behovet for kraftig motor. Etter mange gode konsepter ble det oppdaget at en forflytning av rotasjonsaksen også ville medføre en endring i spindlenes angrepspunkt på ballen. Det ble derfor foretatt noen enkle tester for å undersøke dette nærmere, og det ble konkludert med at dagens rotasjonspunkt gir optimal angrepvinkel på ballen, uavhengig av vuggens vinkelplassering. For å unngå at en eventuell endring ville medføre endringer eller avvik i dagens styringssystem, ble det derfor bestemt å beholde dagens rotasjonspunkt, og heller opplagre dette i bedre grad for å hindre friksjon og redusere kraftbehovet til motoren.

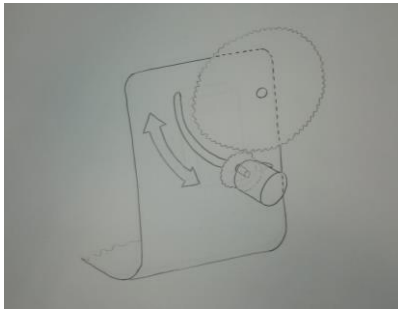
## *Konsepter*

Den enkleste måten å automatisere en mekanisme for å endre vinkelen på vuggen er å sette på en eller annen type motor. For styring av vertikal tilt ble det sett på følgende konsepter.

## Konsept

## Beskrivelse

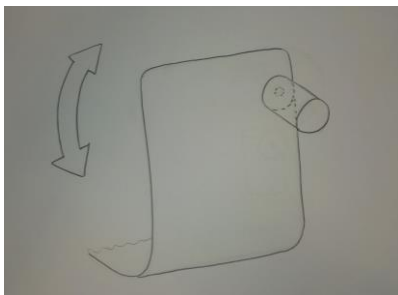
### *Motor følger buebane på siden*



Tanken med dette konseptet var å gjøre færrest mulig inngrep på eksisterende maskin. Et låst tannhjul festes på siden, og et mindre tannhjul festet til vuggen roteres av en motor som vil bevege seg langs buebanen.

Dette var i midlertid en unødvendig komplisert løsning, og motorstørrelse ville her være et vesentlig problem.

### *Motor rett på rotasjonsakse*



En motor rett på rotasjonsaksen er muligens den enkleste løsningen. Med tilgjengelige motoralternativer vil dette dog føre til stor rotasjonshastighet, resulterende i ikke tilstrekkelig dreiemoment og dårlig presisjon.

### *Motor med gir til rotasjonsaksen*



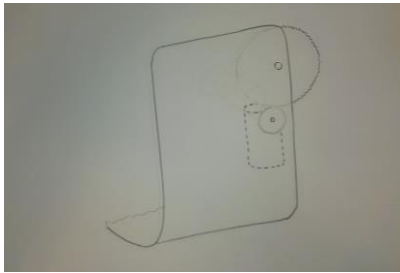
For å løse hastighets- og momentproblematikken kan det implementeres et ekstra gir. Dette er en svært aktuell løsning.

### *Flytte rotasjonsakse til senter, og motor rett på*



Konseptet går ut på å flytte rotasjonsaksen slik at kravet til moment ble redusert. Som nevnt tidligere ville dette få konsekvenser for spindlenes treffpunkt på ballen.

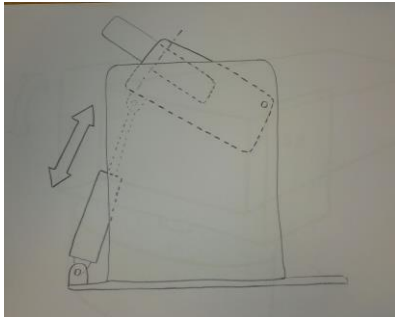
### *Motor med gir på innsiden*



For å holde maskinen kompakt, samt beskytte monterte komponenter mot ev. fotballer, var det ønskelig å plassere motor-/girkøringen på innsiden av maskinen.

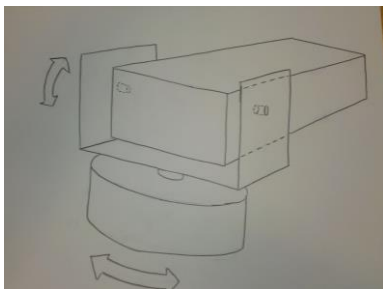
Dette var vanskelig å gjennomføre grunnet plassmangel.

#### *Lineæraktuator foran*



For å unngå kravet om stort dreiemoment ble det vurdert å implementere en lineæraktuator foran, istedenfor en roterende motor på siden. Grunnet plassmangel skulle det vise seg at dette var vanskelig å få til.

#### *Løfte hele maskinen*



En måte å implementere begge aksene i én løsning var å plassere hele maskinen i en rigg som denne. Dette ville dog føre til betydelig større kompleksitet, størrelse og kostnader

**Tabell 1: Konseptutvikling, plassering av rotasjonsakse og motor**

#### *Valgt konsept*

##### *Motor med gir til rotasjonsaksen.*

For å finne ut nødvendig dreiemoment ble det brukt en fjærvekt festet med en viss arm fra rotasjonsaksen. Oppmålingen ga en kraft på 60N og en arm på 180mm. Enkle beregninger med en sikkerhetsfaktor på litt over 2 medførte et behov på 25 Nm dreiemoment.





Det ble oppdaget at det var tegn til mye friksjonsmotstand i opprinnelig rotasjonsmekanisme, noe som resulterer i et behov for høyere torsjonsmoment for vinkelendring, og et lavere behov ved "stand still". Konseptene bygger på å endre opplagringen til bruk av kulelager for å fjerne friksjonsmotstanden og lette arbeidet som må til. Dette vil dog føre til ekstra behov for holdemotstand når ønskelig vinkel er oppnådd og maskinen er i operasjon.

#### *Nødvendige spesifikasjoner for motor*

Strømtilførsel: 12 V batteri  
Dreiemoment:  $\approx 25$  Nm  
RPM: 3 rpm (som blir 0-90° på 5 sekunder)

### Motorløsning

Det ble vurdert en rekke forskjellige motortyper, som i samarbeid med styringssystem og girkjøring kunne gi ønskede egenskaper. Det var i utgangspunktet ønskelig å plassere motor på innsiden av rammen, men det ble ikke funnet motor-gir-løsning som oppfylte kravene til størrelse innenfor det gitte budsjettet.

Type motor	Stepper	Servo	DC (valgt løsning)	AC	Pneumatikk
Nøyaktighet	Høy	Høy	Middels	Middels	Lav
Torsjonsmoment	Svak	Sterk	Sterk	Sterk	Middels
Holdemoment	Svak	Sterk	Middels	Sterk	Svak
Størrelse	Middels	Middels	Middels	Middels	Stor
Styring	Middels	Middels	Enkel	Vanskelig	Enkel
					

Tabell 2: Motorsammenligning

## Girløsning





Det ble raskt oppdaget at for å oppnå ønskelig dreiemoment og rotasjonshastighet var det nødvendig å bruke et eller flere gir. Det første konseptet som ble jobbet med var å bruke et snekkegir for å oppnå stor oversetting, samt selvlåsing slik at holdemomentet ble høyt når ønsket vinkel ble oppnådd. På grunn av lite plass på innsiden av konstruksjonen ble det sett på muligheter for å kombinere dette med to tannhjul for å kunne plassere motoren og snekkegiret lavt, men med et oversettingsledd for å ta rotasjon opp til rotasjonsaksen.

Det viste seg i midlertid at det var svært krevende å finne gode løsninger på motor og snekkegir som var sterke nok for vårt behov og samtidig innenfor budsjettet.

Det ble derfor videre sett på planetgir, som ofte er kompakte og gir stor oversetting, men ikke samme grad av holdemoment. Dette er ofte greit å implementere sammen med motor, men vil gjøre motor-gir-løsningen lengre, noe som var ugunstig på grunn av liten plass.

Det var også ønskelig å finne løsninger som var ferdigbygget i moduler for å begrense tidsbruk og unngå unødvendige komplikasjoner med montering og unøyaktighet. Dette viste seg fort å være dyrt, samt at ønskede krav var svært spesifikke og ikke standard.

Det ble derfor valgt å gå for en standard girløsning med to tannhjul, der gruppen skulle konstruere og bygge girboksen selv. Dette skulle oppfylle kravet til omsetting, men ble av samme grunn noe større enn ønsket. Mye av grunnen var begrensning i budsjettet, da tannhjul koster en god del, så valget falt på to stykk, ett lite (12 mm) og ett stort (120 mm).

	Snekegir	Planetgir	Tannhjul	Lineærløsning
<b>Positiv</b>	<i>Selvlåsende</i>	<i>Stor oversetting</i>	<i>Enkelt</i>	<i>Selvlåsende</i>
<b>Negativ</b>	<i>Mye friksjon</i>	<i>Bygger i lengderetning</i>	<i>Lav oversetting. Stor størrelse</i>	<i>Krever mye plass</i>
				

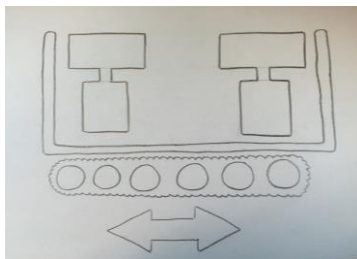
Tabell 3: Girprinsipper

### 2.2.3 Horizontal rotasjon

Prosjektets utgangspunkt var å lage styring av to akser, vertikal og horisontal, men rammene med tanke på både tid og penger satte store begrensninger for planene. Det ble tidlig innsett at bygging og implementering av horisontal rotasjon ikke var reelt innenfor gitte rammer. En horisontal rotasjon vil i de fleste tilfeller gjøre maskinen både større og mer kompleks, og dermed også føre til høye kostnader. Denne delen av oppgaven er derfor håndtert som en mulighetsanalyse som legger opp til potensielt fremtidig arbeid. Her er det blitt evaluert mulige konsepter og løsninger, og styringssystem er dimensjonert og programmert for senere implementering.

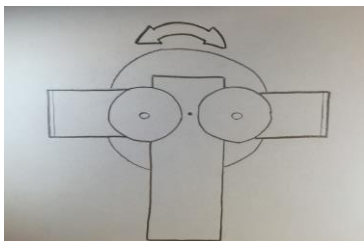
Følgende konsepter ble vurdert for rotasjon i horisontalplanet som vist i tabell 4.

#### Beltevogn



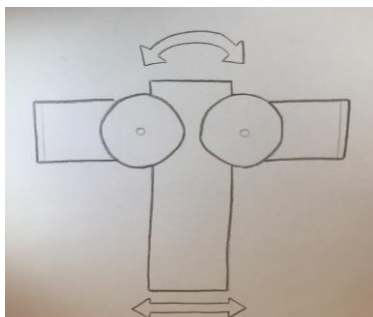
Løsning med belte foran på maskinen som roterer om et rotasjonspunkt bakerst. Stabilt og nøyaktig.

#### Rotasjon av hele maskinen



Løsning hvor hele maskinen står på en plate som roterer, og som igjen hviler på en base som står på bakken. Enkel DC-/steppermotor kan styre rotasjonsplaten ved hjelp av et gir. Solid, men kan kreve en del kraft da maskinen er tung.

#### Rotasjon om utskytningspunkt



Hele maskinen roterer om utskytningspunktet ved hjelp av en hjulmotorløsning bakerst som kan rotere begge veier. Reduserer kraftbehov grunnet lang arm, men kan bli ustabilt.

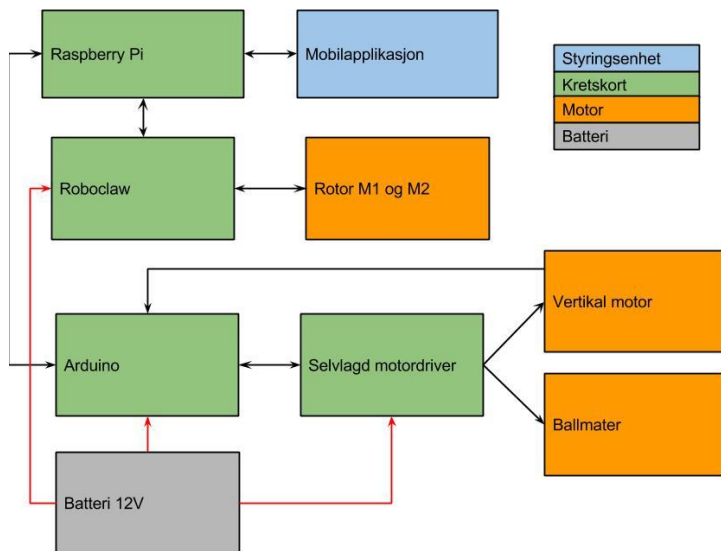
Tabell 4: Prinsipper for horisontalbevegelse

## 2.3 DIGITALT

Her ble det sett på hvordan gruppen ønsket at systemet skulle styres. Gruppen ønsket å kunne styre hele systemet fra én enhet slik at maskinen ble hel-automatisert. Slik maskinen var da den kom, måtte vertikal tilt styres mekanisk for hånd, ballmater ble styrt av et innebygd panel på maskinen mens spindlene ble styrt via PC. Alt dette hadde gruppen som mål at skulle styres via PC/mobilapplikasjon. Dermed måtte gruppen bestemme et styringsdesign og finne de rette komponentene for å få til denne løsningen.

### 2.3.1 Styringskrets

For å kunne styre hele systemet fra én enhet, ble det tenkt at man kunne utvide GUI-en fra masteroppgaven til å inneholde styring av vertikal tilt og ballmaterhastighet. Kretsdesignen som ble tegnet opp som forslag er vist under.



Figur 4: Overordnet styringskrets

Her ble det tatt utgangspunkt i at det hadde vært fordelaktig å kunne styre maskinen fra en mobilapplikasjon, slik at brukeren kunne ha mulighet til å skyte baller til seg selv uten at noen måtte operere maskinen lokalt.

### 2.3.2 Komponenter

Det måtte tenkes ut hvilke komponenter det var bruk for, med den kretsdesignen som ble vedtatt å gå for. Siden maskinen allerede hadde en RoboClaw implementert til styring av spindlene samt programvare (GUI) for å kommunisere med denne, ble det naturlig å beholde og videreutvikle denne løsningen. Det måtte derfor undersøkes hvilke andre komponenter som også trengtes. RoboClaw-en var i utgangspunktet "oppbrukt" på motorspindlene og kunne ikke brukes til noe annet. Siden det skulle styres to motorer til (vertikal tilt og ballmater) trengtes det en løsning for dette.

### *Roboclaw:*

En mulighet var å bruke en RoboClaw til. Denne kan styre både vertikal tilt samt ballmater, da den har to utganger. I tillegg har den inngang til encodere som kunne blitt brukt til vinkel feedback til vertikal tilt.

### *Arduino, motorkontroller og potentiometer:*

Da gruppen hadde problemer med en RoboClaw, ble det også sett på en løsning med en Arduino for å styre en H-bridge/motorkontroller for å styre vertikal tilt og ballmater. Samt å bruke et potentiometer som feedback på vinkelposisjon av vertikal tilt. Feedback er nødvendig da det som tidligere nevnt muligens skulle brukes en DC-motor med gir. DC-motoren er konstruert slik at man ikke kan vite noe om dens posisjon uten å ha en form for måling. Arduinoen er utstyrt med en rekke (analoge og digitale) inn- og utganger, samt seriell kommunikasjon. Det analoge signalet fra potentiometeret kan kobles til en analog-til-digital converter som er innebygget i mikrokontrolleren. Denne gir en oppløsning på 0-1024 (10bit) over 310 grader, som vil bety at vi har oppløsning på 0-297 på 90 grader. Det er altså en målenøyaktighet på  $\pm 0.3$  grader. Dermed har Arduino/potentiometer løsningen mer enn god nok nøyaktighet for vertikal tilt.

#### 2.3.3 Feedback på tiltvinkel

Den mest robuste løsningen som ble vurdert var benyttelse av encodere som baserer seg på bruk av IR-lys eller magnetisme. Disse kan bygges slik at de oppgir absolutt posisjon eller bare gir inkrementell posisjon, altså om motoren flyttet seg et hakk den ene eller andre veien. Encodere er som nevnt en meget driftsikker løsning, men vil kreve en del programmering for å kunne benyttes godt.

Dersom man har små vinkelutslag (0-310 grader) og liten omdreiningshastighet vil et potentiometer kunne benyttes. Dette virker ved at en sirkulært formet motstand legges mellom jord og V+, og at man har en børste som kan flyttes omkring på motstanden slik at spenningen kan varieres mellom jord og V+. Denne løsningen kan være noe ustabil da den lett plukker opp elektromagnetisk støy. I tillegg vil et potentiometer være mindre robust enn en encoder. En positiv egenskap med denne løsningen er at den er absolutt. Den gir alltid ut samme spenning for en gitt vinkel. Dette eliminerer behovet for endestoppbryter ved kalibrering.

#### 2.3.4 Horisontal styring

Ettersom det i starten også var tenkt å implementere horisontal styring, hadde gruppen en idé for styringen av dette også. Gruppen tenkte seg at det kunne lages en ny motordriver (H-bridge), samt å bruke Arduino til kommunikasjon med denne også. Evt. kunne man brukt en RoboClaw for styring av vertikale og horisontale bevegelser, ettersom den har to utganger. Her ville det også vært naturlig og bruke encoderene på RoboClaw-en for feedback-reguleringen. Man ville da uansett trengt både en Arduino og en motorkontroller til ballmateria.



### 2.3.5 Kommunikasjon med bruker

Videre ønsket gruppen å se på hvordan det vil være mest mulig brukervennlig å styre maskinen. Ettersom alt ønskes automatisert, hadde det vært fordelaktig om brukeren kunne styrt maskinen trådløst via en mobiltelefon. Følgende løsninger ble vurdert.

#### *PC:*

Utvidelse av GUI. Her må man plugge til PC og kjøre programmet. Dette er det enkleste å implementere, men også den minst kompakte løsningen. Ved bruk av kun GUI fra PC er en fortsatt nødt til å ha noen til å styre maskinen for seg, ev. kjøre et program som skyter tilfeldig. For å få til trådløs styring, ble det tenkt at man kunne sette opp en server på PC-en som kunne videreformidlet instrukser fra en mobilapplikasjon. Dette ville løst problemet med at man må ha noen til å styre maskinen for seg.

#### *Raspberry pi:*

For å kunne gjøre maskinen mest mulig portabel, ville en potensielt kunne bruke en Raspberry Pi mikrokontroller for kommunikasjon mellom mobilapplikasjon og Arduino/motordrivere. Ved å bruke denne løsningen vil man da kunne slippe å koble til en PC for styring. Dette er den mest elegante løsningen, men innebærer en god del mer jobb med implementering, samt noe ekstra kostand i forbindelse med innkjøp.

#### *Mobilapplikasjon:*

Konseptet rundt mobilapplikasjonen spinner rundt det at den betraktes som en input-enhet. Ønsket oppførsel bestemmes, og dette sendes til datamaskinen/Raspberry Pi som reagerer på dette. Totalt to skjermer ble prioritert der den ene bestemmer kontinuerlig styring eller posisjonering via input i form av eksakte koordinater, og den andre bestemmer selve spinnets på ballen. En tredje skjerm for nettverksinput er også diskutert.

Kommunikasjonen er først og fremst fra mobilapplikasjon til datamaskin, men er åpen for videreutvikling av kommunikasjon motsatt vei også når begge enheter er kompatible med hverandre.

## 3 BEREGNINGER

### 3.1 MOMENT, ROTASJONSHASTIGHET OG GIRING.

Siden det ønskede dreiemomentet som skulle påføres vuggen var på 25 Nm, ble det forsøkt å finne motorløsninger for dette. Det ble raskt konkludert med at det var behov for en gearløsning, slik at man fikk økt momentet og redusert hastigheten fra motoren. Elektriske motorers egenskaper svarer som oftest ikke direkte til behov, og det er derfor vanlig å implementere gir for å justere dette.

Ønsket moment: 25 Nm

Ønsket rotasjonshastighet: ca. 3 rpm (tilsvarende 5 sekunder på 90 grader rotasjon)

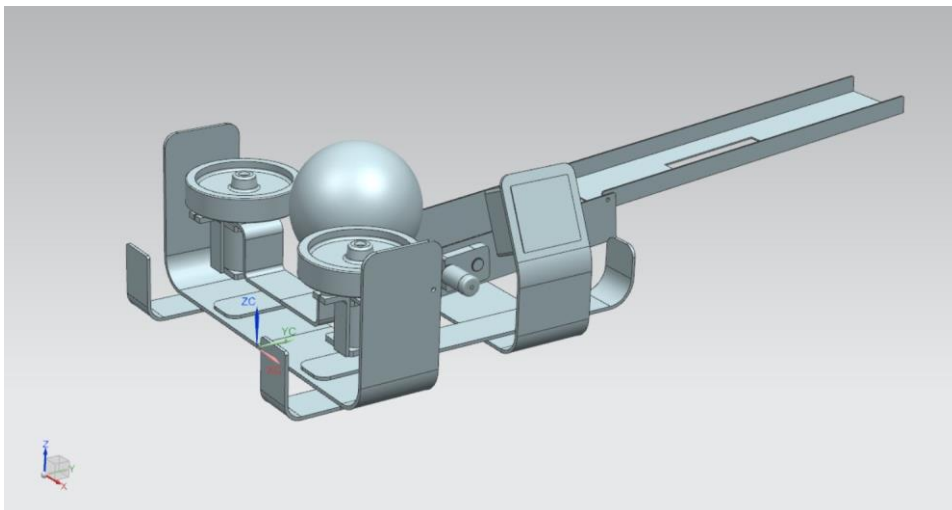
Innkjøpt motor var med ferdig snekkegir-løsning og nedgiring på 55:1. Utgående moment tilsvarte dermed 5 Nm, noe som ikke var tilstrekkelig. For å treffe de gitte kravene måtte det derfor gires ned med ytterligere 1:10.

Resulterende moment som er mulig å påføre vuggen ble derfor på 50 Nm, med en hastighet på 4 rpm. Dreiemomentet ble betydelig høyere enn hva som kreves, men dette var nødvendig for å sikre en lav nok hastighet og påfølgende god nøyaktighet.

### 3.2 CAD OG SIMULERINGER

Ved hjelp av programvare for Computer Aided Design, CAD, ble det laget en 3D-modell av maskinen for å lett kunne prototype og analysere løsninger.

Underveis kunne man plassere komponenter for å undersøke hvordan det kom til å se ut, om det var plass, og hvordan det kunne monteres.

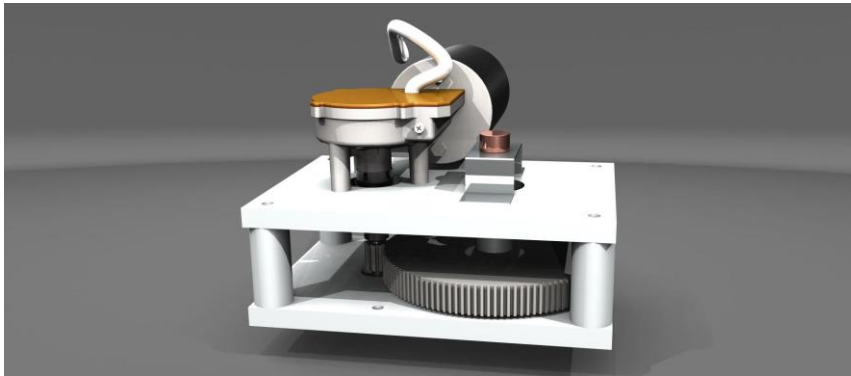


Figur 5: 3D-modell av fotballmaskin

Det ble gjort noen enkle FEM-analyser (Finite Element Method) for å undersøke krefter og bestemme dimensjoner på nye komponenter. Det ble raskt oppdaget at styrke og stivhet ikke var et problem, og fokuset videre ble derfor rettet mot det som var praktisk mtp. produksjon og montasje.

Vektbesparelse og topologioptimalisering var ikke i særlig grad viktig for dette prosjektet, og ble derfor ikke jobbet videre med.

Videre ble det brukt 3D-modellering til å lage komponenter til girkasse for å få ønsket utgangshastighet og moment for å kunne rotere utskytningen. Det ble utarbeidet maskintegninger og illustrasjoner, samt sammensetning av gir, motor og nåværende maskin for å se på plassering av girkasse og motor. Hvordan kraften skulle overføres til nåværende konstruksjon ble også illustrert. Den ferdige konstruksjonen av girkassen ble sendt til produksjon i kybernetikkverkstedet.



Figur 6: 3D-modell av girkasse med motor

### 3.3 PRODUKSJON

Den opprinnelige fotballmaskinen bærer preg av enkle løsninger som lar seg masseprodusere. Det var derfor ønskelig å gjøre en vurdering av produksjonspotensialet ved de nye løsningene.

De nye endringene bærer preg av at man ikke ville gjøre drastiske endringer på det opprinnelige designet. Mest mulig av maskinen er beholdt slik den var. De eneste endringene som er gjort er at den ene akslingen er byttet ut for å implementere opplagring, samt at det har blitt laget noen hull til å feste girboks og koblingsboks. Den største modifikasjonen er girboks med motor som er festet på utsiden. Ved valg av komponenter ble det fokusert på bruk av standardkomponenter, slik at monteringen ble enklest mulig. Girboksen ble designet og bygget "in-house" for å kunne tilpasse dette til tannhjul, motor og fotballmaskin. Den er designet med tanke på produksjonsvennlighet og det som var tilgjengelig ved verkstedet. Verkstedet på kybernetikk har stått for selve produksjonen av girboksen, og kontinuerlig kommunikasjon ble opprettholdt.

Grunnet HMS måtte all maskinering bli gjort av ansatte, og leveringstiden til girboksen ble dermed både lang og usikker.

## 4 LØSNING

Gruppens endelige løsning har resultert i bruk av en DC-motor med snekkegir i kombinasjon med en egenprodusert girboks. Dette er festet til rotasjonsaksen som er opplagret med kulelager for bedre rotasjonsevne og mindre friksjon. Videre blir spindlene, vertikal tilt og ballmateren styrt via en PC der det er laget en GUI skrevet i Python. Mellom PC og spindler er det plassert et RoboClaw-kort, mens det mellom PC og vertikal tilt og ballmater er en Arduino og en egenprodusert H-bridge.

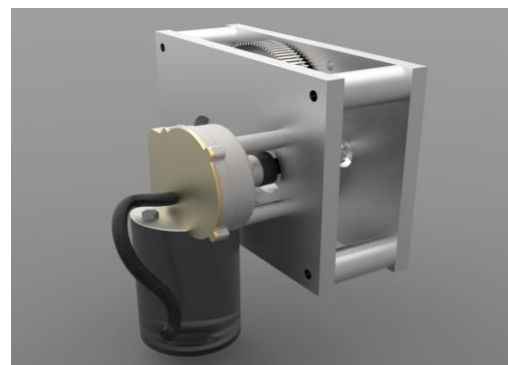
Det er også blitt utviklet en mobilapplikasjon for enkel og brukervennlig trådløs styring. WiFi-kommunikasjonen er ikke implementert, men systemet er oppe og går.

### 4.1 MEKANISK SYSTEM

DC-motor med snekkegir og et utgående dreiemoment på 5 Nm var kjøpt inn og hadde en utgående hastighet på 40 rpm, noe som tilsvarer ca. 0,4 sekunder på 90 grader.

Spenning: 12V  
Effekt: 50W

Tannhjulsgiringen har en overføring på 1:10 som gir en reduksjon på faktor 10 på hastigheten og en tilsvarende økning i moment. Giret er designet med tanke på en lav hastighet. Den lave hastigheten gir bedre nøyaktighet og et moment som er stort nok for å justere og holde ønsket vinkelutslag. Vinkelen skal variere mellom 0 og 90 grader, og med en hastighet på 4 rpm vil det ta omtrent fire sekunder å justere mellom de to ytterpunktene.

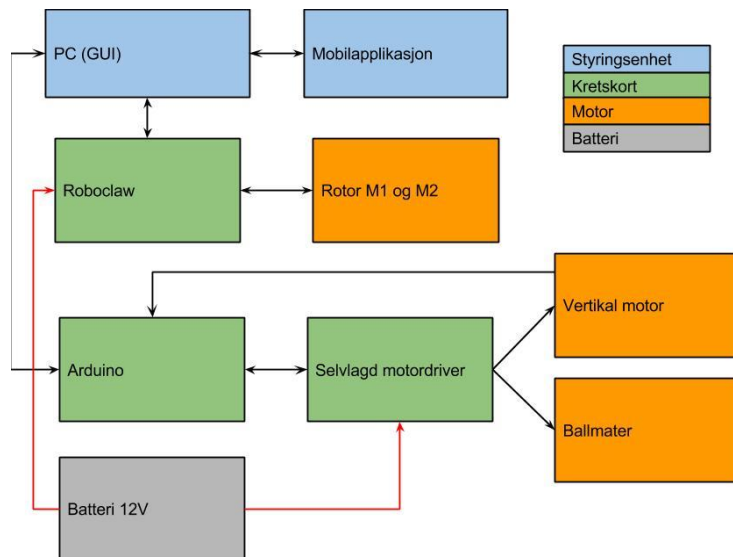


Figur 7: Girboks

### 4.2 MEKATRONISK/ELEKTRONISK SYSTEM

#### *Styringsløsning:*

Gruppen valgte først og fremst å styre maskinen ved å videreutvikle GUI på PC. Gruppen vurderte det som viktigst å få til dette først, før ev. videreutvikling med PC server/Raspberry Pi. Gruppen valgte likevel å lage en mobilapplikasjon slik at den delen var gjort og at kun mellomledet (PC server/Raspberry Pi) gjenstår å implementere. Kretsløsningen gruppen gikk for er vist i figur 8



Figur 8: Overordnet styringskrets, valgt løsning

### Komponentvalg:

Ettersom det var noe problemer med den ene RoboClaw-en gruppen hadde til disposisjon (som senere ble fikset), falt valget av styring av vertikal tilt og ballmater på Arduino og H-bridge-løsningen, med potentiometer som feedback slik som beskrevet tidligere. En P-regulator ble brukt for å regulere vertikal tilt, mens ballmateren ikke behøvde noen regulering.

## 5 IMPLEMENTERING

### 5.1 MEKANISK

#### *Produksjon av deler*

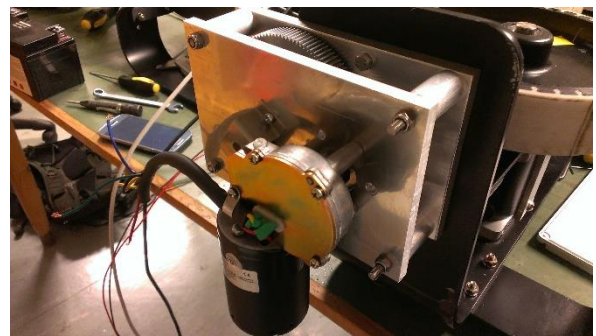
Mesteparten av delene til girboksen er produsert av ansatte ved instituttets verksted. I den anledningen ble det laget to aluminiumsplater med tilpassede hull for kulelager og innfestning. Det er også laget to akslinger og fem «spacere» som sørger for riktig avstand når motoren skal monteres.

#### *Innkjøpte deler*

Motor:	ZD1530L 12V 50W
Motorkontroller:	Roboclaw 2x30amp
Tannhjul:	12 mm og 120 mm
Kulelager:	7 mm og 2x 12 mm (indre diameter)

#### *Montering*

Ettersom monteringen var analysert ved hjelp av CAE-programvare var dette en enkel jobb. Alle mekaniske deler passet sammen som planlagt, med noen små justeringer for å sikre god momentoverføring.



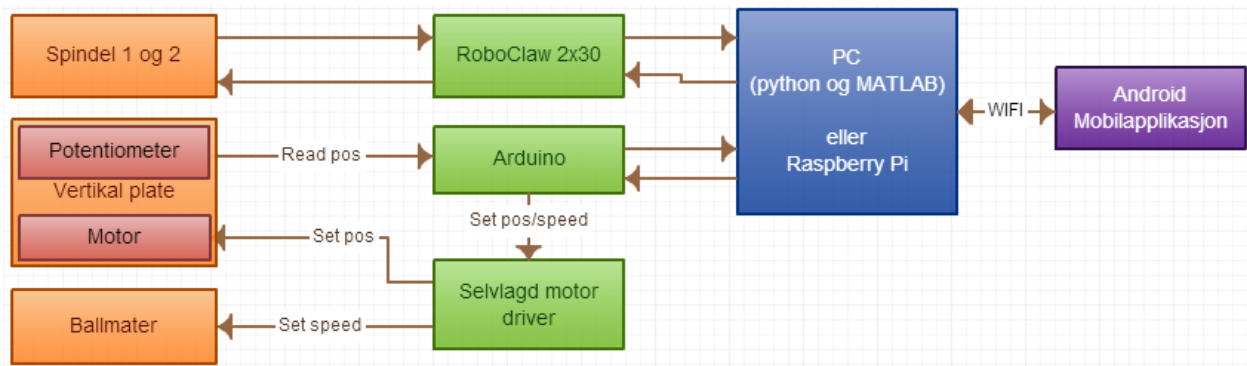
Figur 9: Ferdig montert girboks

### 5.2 DIGITALT (ELEKTRONIKK OG PROGRAMVARE)

Fotballmaskinen var allerede utstyrt med et kretskort for kontroll av motorhastigheten til spindlene. Kortet er et motorkontrollkort av typen RoboClaw 2x30A brushed DC motor controller. Ettersom dette allerede eksisterte, og var blitt implementert av en masterstudent, beholdt gruppen denne konfigurasjonen videre. For å styre vertikal tilt ble det laget en H-bridge motordriver for å drive DC-motoren. En enkel driver for styring av ballmateria ble også implementert på samme printkort. Både H-bridge og driveren til ballmateria ble kontrollert av en Arduino.

## 5.2.1 Styring

### Overordnet styringskrets



Figur 10: Styringskrets med kommunikasjonsforløp

Styringskretsløsningen styrer både de implementasjonene som ble gjort i masteroppgaven, samt implementasjoner av vertikal og horisontal styring av fotballmaskinen og hastigheten på ballmateren. System er tenkt å kunne bli styrt trådløst via en mobilapplikasjon slik at brukeren kan styre maskinen via mobilen på en enkel måte. Systemet vil likevel primært under utvikling og testing bli styrt via USB-kommunikasjon med en PC.

### Styring av vertikal tilt

For å styre DC-motoren til vertikal tilt, designet og konstruerte gruppen en egen motordriver. Denne H-bridge motordriveren blir styrt av en Arduino, og bruker et potensiometer som feedback på vertikal vinkel. Potentiometeret som ble brukt var på 50kΩ. Denne kunne med fordel vært under 10kΩ, men 50kΩ var det vi hadde tilgjengelig. Dette gjør at systemet opplever noe støy. Arduino kommuniserer med PC-en. En enkel regulatoralgoritme er implementert som gir ut pulsbreddemodulerte signaler på to utganger. En utgang til hver retning for bevegelse opp og ned. Det ble forsøkt med både PID-, PI- og PD-regulering, men til slutt fant man ut at en P-regulator fungerte veldig bra med Kp-verdien satt lik 1. Vertikal tilt har gjennom programkoden et vinkelutslag mellom 0 og 80 grader med oppløsning på  $80/255=0.3$  grader som burde være mer enn nok.



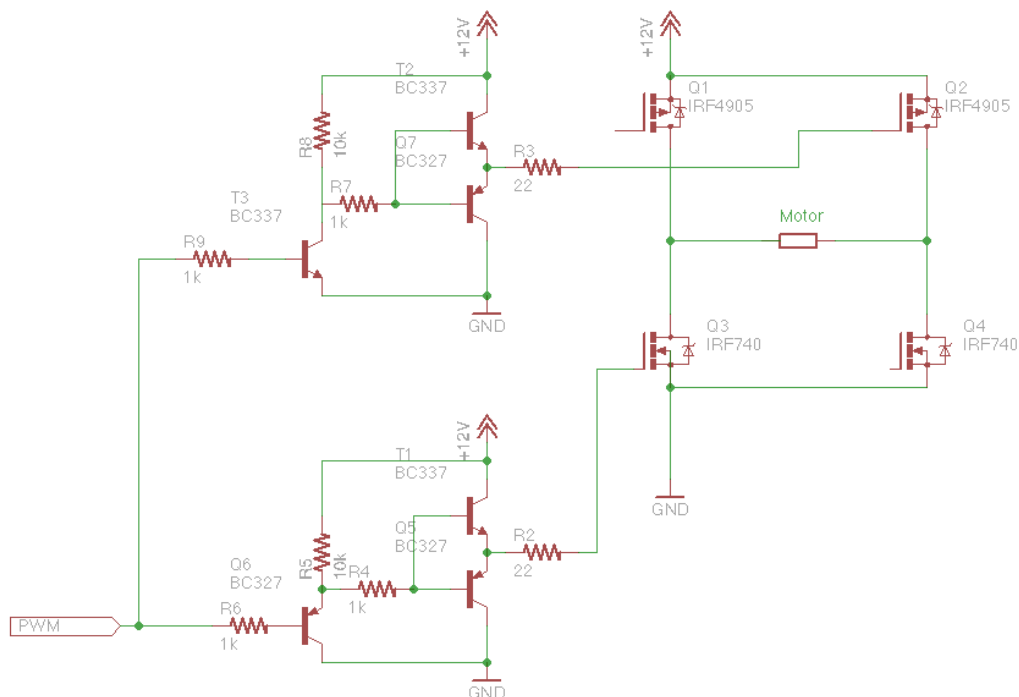
Figur 11: Arduino mikrokontroller

### Styring av ballmater

Her ble det også laget en motordriver, nå med Arduino for kommunikasjon. Det var ikke noe behov for regulering av hastigheten, da det kun justeres spenning mellom 0-12V.

## Motordriver

Ut fra utgangene på Arduino mikrokontrolleren står det montert en motordriver. Denne kan snu polariteten til spenningen inn i motoren og derav snu dreieretningen til motoren for vertikal tilt. Motordriveren er utført som en standard H-bridge, med to N-channel MOSFET (IRF1409) og to P-channel MOSFET (IRF9140). Grunnet stor gate-kapasitans på MOSFET-ene, ble det nødvendig å lage en gate-driver med lav impedans mot henholdsvis GND og +12V for sikre høy spenningsendring ( $dv/dt$ ). En totempole-driver er realisert med to komplementære transistorer (BC337 og BC327) og koblet emitter mot emitter. Basene på disse to er koblet sammen, og er koblet til et forsterkerledd realisert med en transistor som er av typene PNP og NPN for henholdsvis nedre og øvre bein. Dette vil da gjøre at krafttransistorene leder når de skal. For å unngå oscilleringer på gaten har det blitt satt inn 22 ohms motstander mellom driveren og gaten på MOSFET-en. Øvrige motstander er valgt basert på tidligere forsterkerdesign. Dioder i antiparallell med krafttransistorene er nødvendig, da motoren er induktiv og derav vil prøve å opprettholde strømmen gjennom krafttransistorene i det de switches av eller på. Dioder på det komplementære beinet vil da lede og gi strømmen en vei å gå. Ved testing før disse var montert ble det funnet spenningstransienter helt oppe i 300V. Etter montering av diodene ble disse redusert ned til det neglisjerbare. Figuren under viser kretsen med halve driverkretsen. Driverkretsen for de to som ikke er tilkoblet er lik den som er vist.



Figur 12: Driverkrets

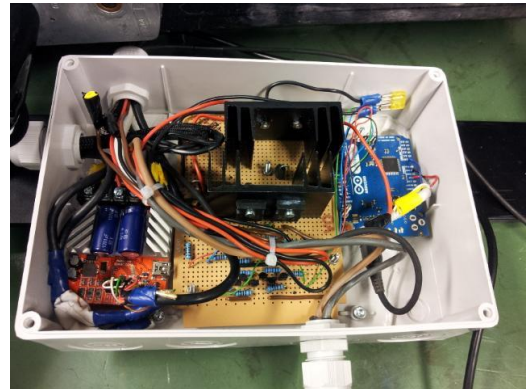
Selve kretsen er loddet på et veroboard da kretsen er forholdsvis enkel. Banene er også forsterket med ekstra kobbertråd der det går høye strømmer. For å holde oppsettet så enkelt som mulig, valgte man å ikke lage et printkort for mikrokontrolleren, men i stedet bruke et utviklingskort med strømforskyning og USB-tilkoblinger som var ferdig. Dette ville også gjøre utførelsen enklere.



## 5.2.2 Oppsett

### Hardware

- Arduino Leonardo mikrokontroller
- Roboclaw motorkontroller til spindler
- Motordriver (H-bridge) til vertikal tilt
- 50k Potentiometer til feedback på tiltvinkelen
- 2 stk. USB (Roboclaw og Arduino)
- Strømkabel (brun = +, svart = -)
- 12V motorsykelbatteri



Figur 13: Koblingsboks

### Roboclaw pin konfigurasjon

Hovedledninger				RoboClaw pins
Seriell			GUL	S2
Seriell	SVART	RØD	ORANSJE	S1
M2 encoder	GRÅ	GRØNN		EN2
M1 encoder	GRØNN	GRÅ		EN1
M2 strøm	HVIT	BRUN		(- +)
M1 strøm	HVIT	BRUN		(- +)

Tabell 5: Roboclaw konfigurasjon. Ledningsoppsett for kommunikasjon med Roboclaw

### Roboclaw-innstillinger

Mode: Packet Serial Mode

Baudrate: 38400bps

Battery options: Battery cut off settings



Figur 14: Roboclaw motorkontroller

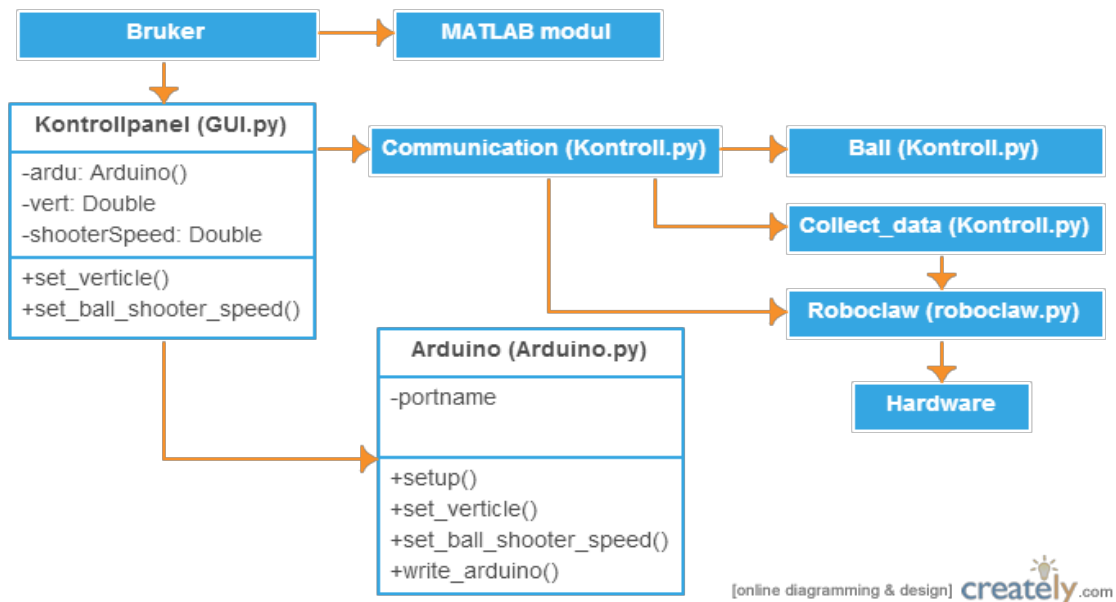
### Arduino pin-konfigurasjon:

P9	PWM for opp/ned vertikal tilt
P10	PWM for opp/ned vertikal tilt
P11	PWM for ballmater
A1	Potentiometer feedback fra vertikal posisjon

Tabell 6: Arduino pin-konfigurasjon

### 5.2.3 Programvare

Python er brukt for å skrive programvaren til systemet, mens MATLAB er brukt i masteroppgaven for beregninger av spinn og hastighet på spindlene. Programstrukturen vil videre forklares her.



Figur 15: Klassediagram

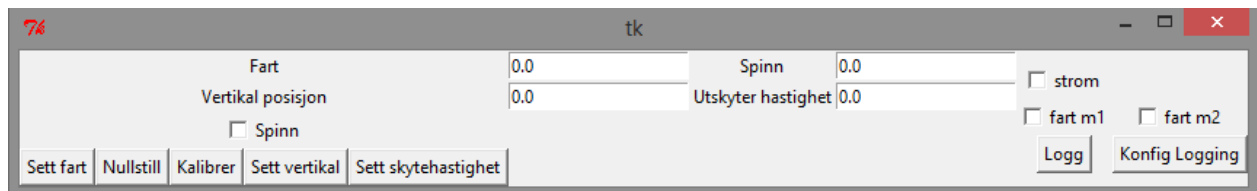
Klassediagrammet viser hvordan programvaren er satt sammen. Helblå bokser var allerede implementert av masterstudent, og det anbefales å lese denne masteroppgaven for mer info om disse modulene.

### Kjøring av program

For å kjøre programmet må man først sjekke at riktige "com"/USB-innganger er satt i koden. "ComX" finnes i Kontroll.py og Arduino.py. Kontroll.py står for kommunikasjon med RoboClaw, mens Arduino.py står for kommunikasjon med Arduino.

### *Kontrollpanel modul (brukergrensesnitt):*

Klassen kontrollpanel har blitt utvidet til å inneholde brukergrensesnitt (GUI) for styring av vertikal tilt og ballmaterhastighet. Da det allerede var laget en GUI til spindelkontroll, ble denne videreutviklet. GUI er skrevet i Python med Tkinter-pakken for det grafiske.



Figur 16: Brukergrensesnitt

### *Arduinomodul:*


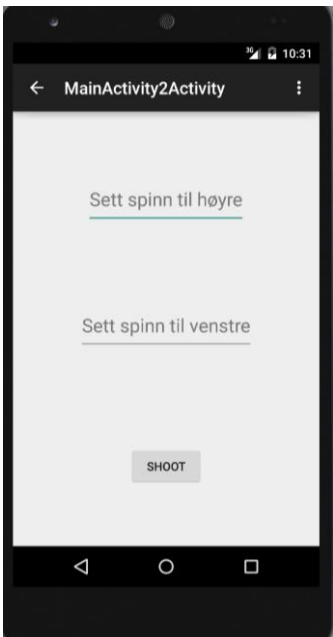
Denne modulen styrer vertikal posisjon. Potentiometer-feedback og P-regulering er satt i Arduino-koden selv. Vertikal tilt kan styres med verdier mellom 0 og 80 grader. Hastigheten på ballmater settes mellom 0 og 100 prosent.

#### 5.2.4 Mobilapplikasjon

Det generelle biblioteket for Android Studio/Java, programmet hvor mobilapplikasjonen ble implementert, og kilder som Youtube og Stackoverflow, ble benyttet aktivt for ideer og generelle tips. Det skulle vise seg at Android Studio støttet et grensesnitt for visuelt design og et programmeringsfelt for ønsket oppførsel.

Konseptet rundt mobilapplikasjon-implementeringen spant litt rundt det å sette opp designet, for så å benytte innebygde funksjonaliteter til å koble designet opp mot den ønskede oppførselen. Det ble også flyttet på knapper og tekstfelt, og kjørt kode for disse objektene hvis det ble detektert aktivitet. Aktivitetene som ble gjennomført gikk ut på å sende og oppdatere status for prosessen, skifte skjermer som vises på mobilen, sende meldinger og holde orden på aktuelle jobber i systemet. Dette ble gjort via funksjoner, implementert i klasser. Kodeteknisk er det ikke en stor oppgave da det for det meste ble benyttet if-setninger i programmeringen. Koplingen mellom designet og koden ble gjort via 'casting' av data- og designtyper.

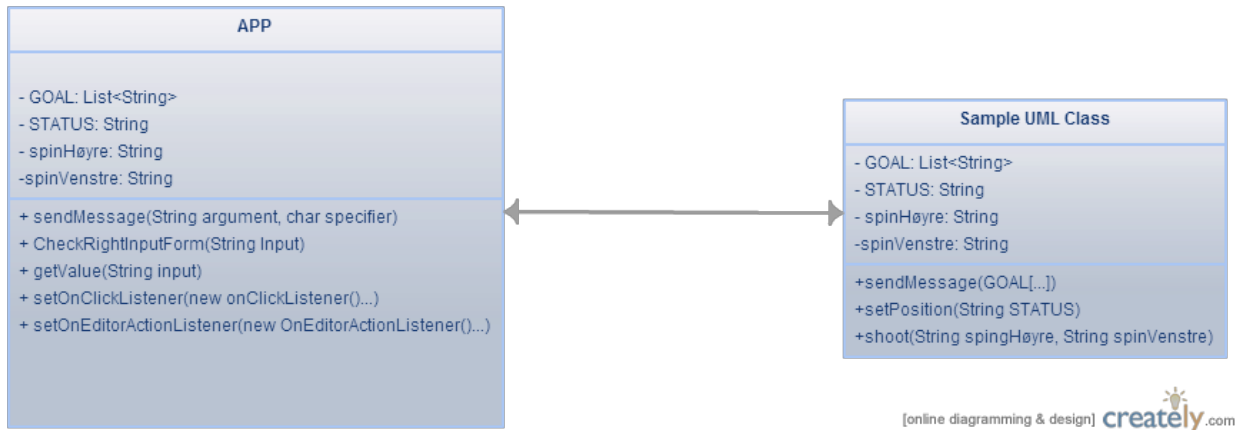
Koden bærer preg av try - catch - statements, da de fleste innebygde funksjoner kaster 'Exceptions'. Dette preger særlig nettverkimplementasjonen. Nettverket bygger på en TCP - tilkopling hvor datamaskinen kjører en server, mens mobilapplikasjonen kobler til som client, når beskjeder skal sendes. Altså, er kun kommunikasjonen fra mobilapplikasjon til datamaskin implementert med mulighet for kommunikasjon den andre veien også.

Skjerm for koordinater og kontinuerlig styring	
	<p>Tanken med denne skjermen er at man enten kan sette prosessen til å styres kontinuerlig med OPP, NED, VENSTRE og HØYRE, eller man kan sette ønsket posisjon på maskinen via koordinattekstfeltet. Har man valgt 'Koordinater', vil maskinen lagre dette som en jobb, og muligheten for kontinuerlig styring vil ikke være mulig før jobben er utført.</p> <p>Man kan på den andre siden stoppe prosessen med STOPP-knapp og maskinen vil da stoppe og alle jobber slettes.</p>
Skjerm for motorstyrke + skytkommando	
	<p>SKYT-knappen på skjerm 1, sender deg til skjerm 2. Her kan man sette styrken på hver av motorene. Dette vil generere en spinn på fotballen. Tanken med skjerm 2 er at maskinen skal vite når den skal være klar til å skyte ved at mobilapplikasjonen sender info når den går inn i skjerm 2 og at den vet når den skal være klar til å forlate skjerm 2 og mulighet for å skyte når den går ut av den.</p> <p>Fotballene skytes når du trykker SHOOT.</p>

Tabell 7: Grensesnitt i mobilapplikasjon

På klassediagrammet under vises en grov oversikt over de viktigste funksjonene i kommunikasjonen mellom mobilapplikasjon og datamaskin. Man kan se for seg en liste, GOAL, som mates med koordinatønsker fra brukeren. Funksjoner som checkRightInputForm sjekker om brukeren har skrevet inn gyldig informasjon der getInput genererer en leselig datamengde ut fra dette. STATUS

oppdaterer seg innenfor tilstander som; kontinuerlig styring (høyre, venstre, opp, ned), og tilstander som SHOOT(skjerm 1 og 2), DONESHOOT(skjerm 2), STOP(skjerm 1).



**Figur 17: Kommunikasjon mellom app og kontroller**

Kommandoene setPosition og shoot på datamaskinsiden utfører de aktuelle operasjonene på maskinen ut fra STATUS, spinHøyre og spinVenstre.

Ideen bak GOAL: List<String> er at alle koordinatjobber brukeren skriver inn lagres i en liste hos begge enheter. Når datamaskinen merker at fotballmaskinen er ferdig med en jobb, vil denne markeres som utført og slettes i listene i både mobilapplikasjon og datamaskin.

## 6 VÅR UTTESTING/RESULTATER

### 6.1 INITIELL TESTING

Før gruppen fikk programvaren til å fungere ble spindlene testet ved å sette strøm direkte på motorene for å se hvordan de fungerte. Ved denne testen var kun halvparten av ønsket strømmengde tilgjengelig, men testen var likevel veldig interessant da den ga en bredere forståelse for hvordan maskinen fungerer i praksis og hvordan kraftoverføring foregår i de mekaniske delene.

### 6.2 FORTLØPENDE TESTING UNDER UTVIKLING

Det tok litt tid å få RoboClaw til å fungere, da det var problemer med innstillingene på kortet samt oppsettet på batteritilførsel. Gruppen tok blant annet kontakt med masterstudenten som også kunne bekrefte at hvordan man slår på systemets batteri har betydning. I tillegg måtte programvaren tilpasses noe til Windows, da den fra før av var implementert i Linux. Gruppen fikk til slutt RoboClaw og programvaren til å fungere. Det viste seg å være nødvendig å koble om to av encoder-ledninger på en av spindlene, da disse var fasevendte og ga negativ rotasjonshastighet.

Første kjøring av maskinen ble utført med enkle kodesnutter. Denne gangen med batteriet som fulgte med og følgelig ønsket strømmengde. Dette ga bedre resultater og man kunne se antydninger til at programmets funksjoner fungerte som det skulle.

Ballmateren gikk veldig fint å implementere. Det var bare å sette inn PWM-verdier og det behøvdes ingen regulering. Da vertikal tilt ble testet ble et potensiometer ødelagt da kommunikasjon mellom GUI og Arduino ikke samstemte helt. Det ble sendt for høye verdier til motoren, som gjorde at motoren gikk for langt og potensiometeret vridde seg. Dette ble løst fort ved å sette på et nytt potensiometer og bedre sikring i koden slik at det ikke ville skje igjen. En ytterligere forbedring vil kunne være å sette inn endestoppbrytere, men dette har ikke blitt gjort. Videre gikk gruppen i gang med reguleringen via et selvlagd test-program. Etter å ha forsøkt PID-, PD- og PI-regulering falt valget til slutt på en enkel P-regulator med Kp satt til 1. Samplingsfrekvensen for denne er 20 Hz. Reguleringen ble da stabil og medførte en presisjon på  $\pm 2$  grader. Gruppen føler dette var en god presisjon for denne maskinen.

### 6.3 TESTING AV GIRLØSNING

Innkjøpt motorløsning ble testet med girkomponentene og fungerte som det skulle. Vuggen var nå blitt opplagret med kulelager på den ene siden, noe som resulterte i betydelig mindre friksjon, og en jevnere rotasjonsbevegelse. Under montasje ble det oppdaget noe mer slark enn ønskelig, og designet ble dermed utbedret ved hjelp av settskruer gjennom akslingene for å sikre god momentoverføring. Sluttresultatet ble betydelig bedre, men ikke helt i henhold til vårt ønske om  $\pm 1$  grad nøyaktighet, da det gjenværende slarket tilsvarer ca. 5 grader. Dette er dog ikke et stort problem under drift, da tyngdekraften gjerne hindrer vuggen i å rotere oppover.

## 6.4 TESTING AV SLUTTRESULTAT

Alle programmets funksjoner, med unntak av wifi-oppkobling mot mobilapplikasjon, er nå på plass og det er blitt gjort en rekke tester for å evaluere sluttproduktet. Alt ser ut til å fungere som forventet. Potentiometerets høye resistans på 50k $\Omega$  gjør systemet noe utsatt for støy, så man må holde fingrene unna potensiometeret for å unngå vinkelfeil. Ellers har systemet god regulering på både spindler og vertikal tilt. Systemet fungerer veldig bra fra den sentrale styringen som gjøres fra PC. Fotballmaskinen er dermed automatisert.

## 6.5 MOBILAPPLIKASJON

Applikasjonen fungerte som den skulle med tanke på generell oppførsel. Informasjonsflyt som skyte, kjøre opp og ned, sending av informasjon mellom to applikasjonsskjermer, ble alt behandlet i riktig rekkefølge. Oppkobling til en ekstern server var også vellykket. Det eneste som ikke ble testet var WiFi-oppkoblingen opp mot datamaskinen. Dvs. testing på maskinsystemet ble ikke gjennomført.

Design delen ble implementert med bruk av enkle predefinerte grafikkløsninger, og mobilkompatibilitet ble ikke undersøkt i større grad. Dvs. hvilke mobilmodeller applikasjonen fungerer på ble ikke testet.

## 7 DISKUSJON

Prosjektet i sin helhet har hatt varierende resultater gjennom hele semesteret. Dette har vært forårsaket av mye motgang i form av ventetid, vanskeligheter med bestilling av deler, og begrenset budsjett. Likevel har det blitt utført godt arbeid forløpende, og dette har resultert i en maskin som fungerer som ønsket. Prosjektets omfang har blitt noe redusert underveis, fra to akser til én, men dette har heller forbedret kvaliteten på det gruppen faktisk har levert.

Girboksen som er blitt laget for denne maskinen er designet for å forenkle produksjon. Den er også dimensjonert opp i stor grad, både for å takle et sterkt holdemoment med mye bevegelse i maskinen fra ballrekyl, samt tåle ev. fotballer som blir skutt på den. Det som kunne blitt gjort videre er å montere på plater (f.eks plexiglass) langs sidene, for å lukke girboksen, og dermed holde den unna regn og støv om maskinen brukes utendørs.

Potentiometeret kunne også vært montert i en tett boks slik at man minsker faren for kortslutninger og målefeil. En kortslutningen mellom noen av pinnene på potentiometeret vil kunne gjøre at tiltmotoren driver vuggen ut av det mulige området og skader vil kunne oppstå.

Bruken av potentiometer kan også diskuteres. Måling vil kunne utføres med inkrementelle eller absolutte encodere som stort sett vil være digitale løsninger som er mindre følsomme for støy. For bruk av inkrementelle encodere er det viktig å lage en løsning som kalibrerer nullpunktet for hver oppstart slik at man alltid vet hvor man er i forhold til nullpunktet for vinkelen. Bruk av RoboClaw motordriver for et slikt system ville kunne blitt vanskeligere da denne ikke har inngang for annet enn inkrementelle encodere og ingen endestopp. Et alternativ ville være å utvide Arduino-koden slik at man kunne bruke encodere. Valget av potentiometer ble gjort på grunn av dens enkle implementasjon og ved bruk av skjermet kabel mellom mikrokontroller og potentiometer fikk man lite støy og en stødig utførelse.

En ordentlig hovedstrømsbryter skulle også vært montert. I og med at det vil kunne være strømmer oppi 40-60 A ved full drift av maskinen bør denne dimensjoneres meget kraftig for å kunne bryte denslags strømmer. Opprinnelig var det montert en lampebryter som man finner ved vanlige vegglamper som normalt sett er designet for å tåle 2-3 A. Denne ble fort varm ved bruk under last. Dette resultatet kan forventes da man potensielt har overbelastet den med 8-10 ganger det den er laget for.

Informasjonsflyten på mobilapplikasjonen baserer seg på tekstfelt der brukeren kan skrive inn ønsket informasjon. Denne informasjonen må være på eksakt form, men dette sjekkes igjen av applikasjonen. Nå sender man f.eks. motorspinninformasjonen enkeltvis før man kan skyte ut ballen uavhengig av input. Dette ble løsningen da det var en sikker og enkel måte å implementere funksjonaliteten.

Wifi-løsningen tok også kun informasjonsflyt fra mobilapplikasjon til datamaskin i betraktning. Dette for å kunne sette opp et enkelt testmiljø for sending av kommandoer. Det ble ikke lagt flere ressurser



inn i kommunikasjonen den andre veien pga. tiden vi hadde til rådighet på slutten. Det samme gjelder den grafiske delen. Uten erfaring, og uten tidsrom for toveiskommunikasjon ble det visuelle lagt litt til side for å sikre en mer robust funksjonalitet.

Gruppen er fornøyde med oppnådd resultat, men ser også både potensielle forbedringer av prosess og løsning. Det er også en rekke ting som kan utbedres og videreutvikles, noe som diskuteres i kapittel 9

## 8 KONKLUSJON

Gruppen startet ut noe ambisiøse med et ønske om å implementere ferdig styring og automatisering av to roterende akser, samt en mobilapplikasjon for styring av hele systemet til slutt. De roterende aksene dreide seg om vertikal tilt og horisontal rotasjon.

Vertikal tilt er i all hovedsak fullført. Det ble vurdert flere løsninger for å utføre dette, og etter litt diskusjon kom gruppen fram til den valgte løsningen med motor og gir. Denne løsningen fungerer, og styring via PC er på plass og i henhold til den standarden gruppen ønsket fra starten av prosjektet.

Horisontal rotasjon er ikke implementert, men et utviklingsgrunnlag er lagt ned for senere arbeid. Det er i den anledning laget konsepttegninger som viser tenkte måter å implementere rotasjonen på.

Styringsmekanismen fungerer som den skal, og alt er samlet til å kunne styres fra én enhet. Målet om å gjøre maskinen hel-automatisert er gjennomført med gode resultater på reguleringen.

Det er tydelig at mange av komplikasjonene og utfordringene som har oppstått har vært i forbindelse med tid og ressurser. Valgt motor- og girløsning har blitt noe annerledes enn hva som ville vært optimalt, men den valgte løsningen kan ses på som en god prototype og "proof of concept". Elektronikksiden har også blitt litt annerledes enn det som var optimalt. Dette kom i form av en egenprodusert krets heller enn et ferdig motorkontrollkort, grunnet problemer med å få kontrollkortet til å kommunisere. I tillegg ble det brukt et potentiometer i stedet for en encoder for feedback på tiltvinkel. Gruppen forsøkte dermed å spare tid ved å gi opp løsninger som kunne ta for mye tid å få implementert.

Sett bort fra litt støy fra potentiometeret ser løsningen ut til å fungere glimrende. Rotasjon av vertikal tilt beveger seg jevnt og stødig, og snekkegiret er i stand til å opprettholde tilstrekkelig holdemoment, også under rekylopptak. Gruppens mål var en nøyaktighet på  $\pm 1$  grad for vertikal tilt. Sluttnøyaktigheten endte på  $\pm 2$  grader, blant annet grunnet toleranser under produksjon, og noen komplikasjoner under testing som resulterte i litt slark i mekanismen. Alt av programvare fra tidligere oppgave er også implementert i et felles brukergrensesnitt og fungerer utmerket.

Med den tiden og de ressursene gruppen hadde til rådighet i dette prosjektet er gruppen svært fornøyd med resultatet. Prosjektet er gjennomført med mye motgang fra utvendige kilder, men med god motivasjon innad i gruppen. Etter reduksjon av oppgavens omfang ble resultatene også i mye større grad kvalitetssikret. Det som ble utført fungerte godt, og sett i en større sammenheng kan det dermed konkluderes med at prosjektet er godt gjennomført, og i henhold til de standardene som gruppen forventet etter reduksjon av oppgavens omfang.

## 9 VIDERE ARBEID

I og med at den opprinnelige oppgaven ble redusert i omfang gjenstår det derfor en del arbeid for å blant annet implementere horisontal rotasjon. Det digitale styringssystemet er allerede designet for å kunne implementere en slik oppgradering, men det gjenstår mye innen mekanisk design. Det er beskrevet flere mulige konsepter for mekanisk design under Konseptstudie.

Styringssystemet og mobilapplikasjonen fungerer godt hver for seg, men det mangler noe arbeid for å opprette en WiFi-forbindelse mellom disse, og dermed tillate fjernstyring. Med dagens løsning vil en kunne sette opp datamaskinen som server og mobilapplikasjonen som klient. Videre kan systemet implementeres med en Raspberry Pi mikrokontroller for å eliminere behovet for en ekstern datamaskin som tar seg av beregninger og kommunikasjon med Arduino og Roboclaw på den ene siden og mobilapplikasjonen på den andre. Brukergrensesnittet kan også forbedres.

Brukergrensesnittet og layout på mobilapplikasjonen kan også videreutvikles om ønskelig, med større knapper og mer brukervennlig design for små mobilskjermer. Mobilapplikasjonen bør testes på forskjellige enheter for å kartlegge hvilke mobiltelefonmodeller den fungerer på. Her kan det hende skalering og kodekomplabilitet er begrensende. Flere funksjoner kan også evt. legges til, om maskinen får utvidet funksjoner.

Det bør implementeres en endestoppbryter på maskinen for den vertikale tilten. Dette er en sikkerhetsbarriere som stopper motorens strømtilførsel når vuggen beveger seg over eller under sitt arbeidsområde. Det er viktig å forhindre dette scenarioet, da det kan medføre vridning av potentiometer, i tillegg til en betydelig strømkning til motoren dersom den forsøker å bevege på akslingen når denne holdes fast. Sistnevnte vil typisk skje når vuggen stanger mot en annen maskindel.

Alternative funksjoner som kan vurderes er nevnt under. Det er beskrevet hvordan disse er tenkt å brukes. Derimot er de svært eksperimentelle, og det er derfor ikke beskrevet eller tenkt ut noen løsning for implementasjon av dem med dagens system.

### *Tilfeldig utskytningsfunksjon*

Maskinen velger en tilfeldig vinkel å skyte fra, kan brukes for å overraske.

### *Sonestyrte treff*

Skal kunne skyte med variert styrke og posisjon innenfor avgrensede områder.

### *Håndbevegelser for styring*

Brukes til styring istedenfor mobilapplikasjon, svært eksperimentelt.

### *Lydanlegg for tilbakemelding*

For moroskyld for det meste.

## *Scoreboard*

For å se om keeper klarer redde ball eller ikke.

## *Kinect*

Keepertrening: Fotballmaskinen ser mål og keeper. Prøver å skyte forskjellige steder i målet.

Pasningstrening: Fotballmaskinen ser spiller, beregner og skyter ballen til spiller.